IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Group Art Unit: 2124
Examiner: Unassigned

In Re PATENT APPLICATION Of:

Applicants	: Kimito HORIE	)
Serial No.	: 10/648,373	) **SUBMISSION OF**
Filed	: August 27, 2003	) **PRIORITY DOCUMENT**
For	: INTEGRATED CIRCUIT	)
Attorney Ref.	: OKI 372	)

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Submitted herewith is a certified copy of applicant's first-filed Japanese Application No. 2002-246629, filed August 27, 2002, the rights of priority claimed on August 27, 2003 with the filing of the above-referenced application.

It is respectfully requested that receipt of this priority document be acknowledged.

FEE ENCLOSED:$ 0
Please charge any further fee to our Deposit Account No. 18-0002

Respectfully submitted,

Robert Berdo

December 17, 2004
Date

Robert H. Berdo, Jr. (Reg. No. 38,075)
RABIN & BERDO, P.C.
(Customer No. 23995)
Telephone: (202) 371-8976
Telefax: (202) 408-0924

RHB:pjl

日　本　国　特　許　庁
JAPAN PATENT OFFICE

CERTIFIED COPY OF PRIORITY DOCUMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application: ２００２年　８月２７日

出願番号
Application Number: 特願２００２－２４６６２９

ST.10/C]: ［ＪＰ２００２－２４６６２９］

出願人
Applicant(s): 沖電気工業株式会社

２００３年　２月１４日

特許庁長官
Commissioner,
Japan Patent Office 太田信一郎

出証番号　出証特２００３－３００６９４７

【書類名】　特許願

【整理番号】　KT000450

【提出日】　平成14年 8月27日

【あて先】　特許庁長官　太田　信一郎　殿

【国際特許分類】　G06F　7/50

【発明者】

【住所又は居所】　東京都港区虎ノ門１丁目７番１２号　沖電気工業株式会社内

【氏名】　堀江　公人

【特許出願人】

【識別番号】　000000295

【氏名又は名称】　沖電気工業株式会社

【代理人】

【識別番号】　100095957

【弁理士】

【氏名又は名称】　亀谷　美明

【電話番号】　03-5919-3808

【選任した代理人】

【識別番号】　100096389

【弁理士】

【氏名又は名称】　金本　哲男

【電話番号】　03-3226-6631

【選任した代理人】

【識別番号】　100101557

【弁理士】

【氏名又は名称】　萩原　康司

【電話番号】　03-3226-6631

【手数料の表示】

【予納台帳番号】　040224

【納付金額】　　21,000円

【提出物件の目録】

【物件名】　　明細書　1

【物件名】　　図面　1

【物件名】　　要約書　1

【包括委任状番号】　9707549

【包括委任状番号】　9707550

【包括委任状番号】　9707551

【プルーフの要否】　　要

【書類名】　明細書

【発明の名称】　加算器，乗算器，及び集積回路

【特許請求の範囲】

【請求項１】　第１と第２の２つのデータ入力と，第１と第２の２つのキャリ入力と，キャリ選択入力とを入力とし，

前記第１と第２の２つのデータ入力のＸＯＲ出力を生成する第１のＸＯＲ素子と，

前記ＸＯＲ出力を選択信号として前記第１のキャリ入力と前記第１のデータ入力とのいずれかを選択する第１のマルチプレクサと，

前記第２のキャリ入力と前記第２のデータ入力とのいずれかを選択する第２のマルチプレクサと，

前記キャリ選択入力を選択信号として前記第１と第２の２つのキャリ入力のいずれかを選択する第３のマルチプレクサと，

前記第３のマルチプレクサの出力と前記ＸＯＲ出力とのＸＯＲ出力を生成する第２のＸＯＲ素子とを有し，

前記第１のマルチプレクサの出力を第１のキャリ出力とし，

前記第２のマルチプレクサの出力を第２のキャリ出力とし，

前記第３のマルチプレクサの出力を加算値としたことを特徴とする加算器。

【請求項２】　請求項１に記載の加算器が複数段継続接続され，

前段の第１のキャリ出力を後段の第１のキャリ入力とし，

前段の第２のキャリ出力を後段の第２のキャリ入力とし，

前記複数の加算器のキャリ選択入力を全段で共通とし，

初段の加算器の真のキャリ入力を前記キャリ選択入力とし，第１のキャリ入力を第１の仮想キャリとし，第２のキャリ入力を第２の仮想キャリとし，

前記キャリ選択入力により最終段の加算器の第１と第２の２つのキャリ出力のいずれかを選択する第４のマルチプレクサを有し，

前記第４のマルチプレクサの出力をキャリ出力としたことを特徴とする加算器。

【請求項３】　前記初段の加算器の前記第３のマルチプレクサを省き，前記

初段の加算器の前記第３のマルチプレクサの出力の替わりに前記キャリ選択入力を用いることを特徴とする請求項２に記載の加算器。

【請求項４】　請求項２または請求項３のうちのいずれか１項に記載の加算器を複数個継続接続し，前段のキャリ出力を後段のキャリ入力とする構成の加算器であって，

最終段の加算器を除き，各段の加算器の処理可能なビット数は，その１つ前段の加算器の処理可能なビット数と等しいか，または大きくしたことを特徴とする加算器。

【請求項５】　初段の加算器を除き，各段の加算器の処理可能なビット数と，その１つ前段の加算器の処理可能なビット数との差が一定であることを特徴とする請求項４に記載の加算器。

【請求項６】　ツリー構造で構成されるブース乗算器であって，最終段の加算器に，請求項４または請求項５に記載の加算器を用いたことを特徴とするブース乗算器。

【請求項７】　請求項１，２，３，４，または５のうちのいずれか１項に記載の加算器を含む集積回路。

【請求項８】　請求項６に記載の乗算器を含む集積回路。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は，加算器，乗算器，及びこれらのいずれかを用いた集積回路に関し，特に，従来のキャリセレクト方式を改良して回路量を減らし，かつ，高速化した加算器，乗算器，及びこれを用いた集積回路に関する。

【０００２】

【従来の技術】

近年のＤＳＰ（Ｄｉｇｉｔａｌ　Ｓｉｇｎａｌ　Ｐｒｏｃｅｓｓｉｎｇ）の発達はめざましく，その中核技術として３２ビット乗算器等が採用されている。その構成には全加算器をN^2個用いた単純な乗算器は採用されず，多くは２次ブース乗算器（Ｂｏｏｔｈ　Ｍｕｌｔｉｐｌｉｅｒ）が採用される。加算器を経由す

るキャリ（Ｃａｒｒｙ）の遅れが大きいことから，部分積（ブース積；Ｂｏｏｔｈ　Ｍｕｘｅｓ）の個数を減らして，回路量の削減や乗算速度の向上を追求するためである。部分積の寄せ集めについては，ＣＰＡ（Ｃａｒｒｙ　Ｐｒｏｐａｇａｔｉｏｎ　Ａｄｄｅｒ）を用いる方法もあるが，キャリの遅れを考慮して，ワレスツリー（Ｗａｌｌａｃｅ　Ｔｒｅｅ）を採用することが多い。

【０００３】

ワレスツリーは，ＣＳＡ（Ｃａｒｒｙ　Ｓａｖｅ　Ａｄｄｅｒ）のツリー構造で構成される。ＣＳＡはキャリ別加算を行うのでキャリの伝搬が起こらず，キャリの遅延も少なくて済む。しかし，寄せ集めたブース積は最終段で６５ビット加算器により通常の加算が行われるので，その部分のキャリの遅れが乗算器の高速化のネックになっていた。また，６４ビット乗算器においても１２９ビットの高速加算器が要求されるため，このことが６４ビット乗算器を実現する際の障害になっていた。

【０００４】

この事情は暗号の分野では特に深刻である。暗号の秘匿性を向上させるためには多ビットの加減乗除を行う必要があり，場合により剰余演算も必要である。今日では，ＲＳＡ方式の暗号では１０２４ビットの数を，楕円曲線暗号方式では２２４ビットの数を扱う必要がある。暗号を生成し，若しくは復号し，更に署名やその検証を行うためには，このような多ビットの計算を高速で行わなければならない。

【０００５】

加算器を高速化する方法には様々なものが知られている。なかでも，キャリルックアヘッド方式加算器（Ｃａｒｒｙ　Ｌｏｏｋ－Ａｈｅａｄ　Ａｄｄｅｒ），キャリスキップ方式加算器（Ｃａｒｒｙ－Ｓｋｉｐ　Ａｄｄｅｒ），及び，キャリセレクト方式加算器（Ｃａｒｒｙ－ｓｅｌｅｃｔ　Ａｄｄｅｒ）がよく知られている。これらの方式は，いずれも加算器を経由するキャリの遅れが大きいことに着目し，これを高速化するアイデアを採用している。このうち「キャリセレクト方式加算器」は，従来あまり採用されていなかった。この方式では，回路量が増大してＬＳＩのコストが高くなり，多ビットの加算器に向いていなかったから

である。

【０００６】

従来の４ビットのキャリセレクト方式加算器は，仮想キャリＶＣ$_0$を入力とする４ビット加算器と，仮想キャリＶＣ$_1$を入力とする４ビット加算器と，これら２つの４ビット加算器のそれぞれの加算値Ｓ$_1$とＳ$_2$とを選択する第１のマルチプレクサと，これら２つの４ビット加算器のそれぞれのキャリ出力Ｃ$_1$とＣ$_2$とを選択する第２のマルチプレクサとで構成される。

【０００７】

２つの４ビット加算器の構成については特に制約はなく，上記キャリルックアヘッド方式加算器等を採用することができる。仮想キャリＶＣ$_0$は‘０’，仮想キャリＶＣ$_1$は‘１’と定めることができるが，その逆であっても良い。

【０００８】

この従来のキャリセレクト方式加算器では，真のキャリ入力Ｃ$_{in}$を第１，第２のマルチプレクサの選択信号とし，真の加算値Ｓ，若しくはキャリ出力Ｃ$_{out}$を選択する。２つの４ビット加算器は，いずれも４ビット入力Ｐ及びＱを共通にするので，必ず加算値Ｓ$_1$若しくはＳ$_2$のいずれかが真の加算値Ｓになり，対応するキャリ出力Ｃ$_1$若しくはＣ$_2$のいずれかが真のキャリ出力Ｃ$_{out}$になる。

【０００９】

このような構成の従来のキャリセレクト方式加算器は，計算速度の面では大きな利点を有している。予め入力Ｐ及びＱが確定していると，この加算器を継続接続した場合には，ほぼ同時に計算が終了する。このため，その後は真のキャリを選択するだけで済み，加算に要する計算時間を節約できる。即ち，キャリの伝搬を考慮しなくてもよい。

【００１０】

【発明が解決しようとする課題】

しかし，従来のキャリセレクト方式加算器は，単なる４ビット加算器と比べると２倍以上の回路量を必要とした。

【００１１】

従来のキャリセレクト方式加算器で使用される２つの全加算器（Ｆｕｌｌ　Ａｄｄｅｒ）は，同一の入力値$Ｐ_{k}$及び$Ｑ_{k}$を入力し，それぞれ加算値$Ｓ_{1}$または$Ｓ_{2}$及びキャリ$Ｃ_{out1}$または$Ｃ_{out2}$を出力する。加算の際のキャリ入力は，$Ｃ_{in1}$または$Ｃ_{in2}$であって，上記キャリ出力$Ｃ_{out1}$または$Ｃ_{out2}$に対応して互いに独立したキャリの経路を作る。そして，マルチプレクサにより，上記加算値$Ｓ_{1}$または$Ｓ_{2}$のいずれかをキャリ選択信号Ｃｓに従って選択し，加算値$Ｓ_{k}$として出力する。このような従来のキャリセレクト方式加算器の回路は，全加算器の加算後に，上記マルチプレクサでそれらの加算値を選択するので，回路の無駄が多かった。

【００１２】

そこで，本発明の第１の目的は，従来のキャリセレクト方式加算器を改良して，必要とする回路量を減らし，多ビット乗算器や暗号技術などにおける多ビットの加算に最適な回路構成を提供することにある。また，本発明の第２の目的は，従来の単純なキャリセレクト方式加算器を改良してより高速な加算器を提供することにある。

【００１３】

【課題を解決するための手段】

本発明の第１の目的を達成するために，本発明では加算値の選択をそれぞれの加算器で加算した後に行うのではなく，先に選択したキャリを使って加算する方式を採用し，併せて共通する回路を削減する。

【００１４】

また，本発明の第２の目的を達成するために，本発明では「適応キャリセレクト方式（Ａｄａｐｔｉｖｅ　Ｃａｒｒｙ　Ｓｅｌｅｃｔ　Ａｄｄｅｒ）」と呼ぶ新しい方式を採用し，多ビット値の高速加算を達成する。

【００１５】

即ち，上記目的を達成するための本発明の加算器は，第１と第２の２つのデータ入力と，第１と第２の２つのキャリ入力と，キャリ選択入力とを入力とし，第１と第２の２つのデータ入力のＸＯＲ出力を生成する第１のＸＯＲ素子と，このＸＯＲ出力を選択信号として第１のキャリ入力と第１のデータ入力とのいずれか

を選択する第１のマルチプレクサと，第２のキャリ入力と第２のデータ入力とのいずれかを選択する第２のマルチプレクサと，キャリ選択入力を選択信号として第１と第２の２つのキャリ入力のいずれかを選択する第３のマルチプレクサと，第３のマルチプレクサの出力とＸＯＲ出力とのＸＯＲ出力を生成する第２のＸＯＲ素子とを有する。そして，第１のマルチプレクサの出力を第１のキャリ出力とし，第２のマルチプレクサの出力を第２のキャリ出力とし，第３のマルチプレクサの出力を加算値としたことを特徴とする。

【００１６】

このような構成により，従来のキャリセレクト方式加算器よりも必要とする回路量（ゲート数）が少なく，しかも演算速度が高速なキャリセレクト方式の加算器が得られる。

【００１７】

また，上記目的を達成するための本発明の他の加算器は，上記構成の加算器が複数段継続接続され，前段の第１のキャリ出力を後段の第１のキャリ入力とし，前段の第２のキャリ出力を後段の第２のキャリ入力とし，複数の加算器のキャリ選択入力を全段で共通とし，初段の加算器の真のキャリ入力をキャリ選択入力とし，第１のキャリ入力を第１の仮想キャリとし，第２のキャリ入力を第２の仮想キャリとしている。そして，キャリ選択入力により最終段の加算器の第１と第２の２つのキャリ出力のいずれかを選択する第４のマルチプレクサを有し，第４のマルチプレクサの出力をキャリ出力としたことを特徴とする。

【００１８】

上記構成では，初段の加算器の第３のマルチプレクサを省き，初段の加算器の第３のマルチプレクサの出力の替わりにキャリ選択入力を用いることができる。

【００１９】

また，上記目的を達成するための本発明の他の加算器は，上記いずれかの構成の加算器を複数個継続接続し，前段のキャリ出力を後段のキャリ入力とする構成の加算器であって，最終段の加算器を除き，各段の加算器の処理可能なビット数は，その１つ前段の加算器の処理可能なビット数と等しいか，または大きくしたことを特徴とする。

【００２０】

上記構成の加算器では，初段の加算器を除き，各段の加算器の処理可能なビット数と，その１つ前段の加算器の処理可能なビット数との差を一定にすることが望ましい。

【００２１】

また，上記目的を達成するための本発明のブース乗算器は，ツリー構造で構成されるブース乗算器であって，最終段の加算器に，上記複数段の構成の加算器のうちのいずれかを用いたことを特徴とする。

【００２２】

【発明の実施の形態】

以下に，本発明のいくつかの実施の形態を，図面を用いて説明する。なお，本明細書及び図面において，実質的に同一の機能構成を有する構成要素については，同一の符号を付することにより重複説明を省略する。

【００２３】

図１は，本発明の第１の実施の形態によるキャリセレクト方式加算器の１ビット分の構成を示すブロック回路図である。以下，図１に示す構成単位を，ＲＣＳＡ（Ｒｅｄｕｃｅｄ　Ｃａｒｒｙ　Ｓｅｌｅｃｔ　Ａｄｄｅｒ）という。

【００２４】

本発明の第１の実施の形態によるキャリセレクト方式加算器（ＲＣＳＡ）（１００）は，第１と第２の２つのデータ入力（P_k，Q_k）と，第１と第２の２つのキャリ入力（C_{in1}，C_{in2}）と，キャリ選択入力（C_s）とを入力とする。

【００２５】

そして，第１と第２の２つのデータ入力（P_k，Q_k）のＸＯＲ出力（$P_k \oplus Q_k$）を生成する第１のＸＯＲ素子（XOR_1）（１０２）と，このＸＯＲ出力（$P_k \oplus Q_k$）を選択信号として第１のキャリ入力（C_{in1}）と第１のデータ入力（P_k）とのいずれかを選択する第１のマルチプレクサ（MPX_1）（１０４）と，第２のキャリ入力（C_{in2}）と第２のデータ入力（Q_k）とのいずれかを選択する第２のマルチプレクサ（MPX_2）（１０５）と，キャリ選択入力

（C_s）を選択信号として第１と第２の２つのキャリ入力（C_{in1}，C_{in2}）のいずれかを選択する第３のマルチプレクサ（MPX_3）（１０１）と，第３のマルチプレクサ（MPX_3）（１０１）の出力（C_k）とXOR出力（P_k@Q_k）とのXOR出力を生成する第２のXOR素子（XOR_2）（１０３）とを有する。

【００２６】

そして，第１のマルチプレクサ（MPX_1）（１０４）の出力を第１のキャリ出力（C_{out1}）とし，第２のマルチプレクサ（MPX_2）（１０５）の出力を第２のキャリ出力（C_{out2}）とし，第３のマルチプレクサ（MPX_3）（１０１）の出力を加算値（S_k）とする（図１）。なお，ここで記号「@」は，XOR（eXclusive　OR）論理演算子である。

【００２７】

図１に示す本発明の第１の実施の形態によるキャリセレクト方式加算器RCSA（１００）では，先にキャリを選択し，その後，選択されたキャリを用いて加算値を計算する方式を採用する。図１に示すように，素子XOR_1（１０２）によって出力される，入力P_kと入力Q_kとの排他論理和P_k@Q_kは，第１のマルチプレクサMPX_1（１０４）と第２のマルチプレクサMPX_2（１０５）のキャリ選択信号として機能する。他方，第３のマルチプレクサMPX_3（１０１）は，キャリ入力C_{in1}またはC_{in2}のいずれかを選択し，その選択には，キャリ選択信号C_sが用いられる。

【００２８】

選択されたキャリC_kは，上記出力P_k@Q_kと共に素子XOR_2（１０３）への入力となり，当該ビットにおける加算値S_kを出力する。また，第１のマルチプレクサMPX_1（１０４）は，キャリ入力C_{in1}または入力Q_kのいずれかを選択してキャリC_{out1}として出力する。そして，第２のマルチプレクサMPX_2（１０５）は，キャリ入力C_{in2}または入力Q_kのうちのいずれかを選択してキャリC_{out2}として出力する。

【００２９】

論理記号で考察した場合，通常，加算値SとキャリC_{out}との関係は，

$S=P@Q@C_{in}$　　・・・　式１

$C_{out}=PQ+QC_{in}+PC_{in}$　　・・・　式２

と表現される。しかし，C_{out}については上記の式２ではなく，次式の，

$C_{out}=C_{in}(P@Q)+P(P@Q)\_$　　・・・　式３

若しくは，

$C_{out}=C_{in}(P@Q)+Q(P@Q)\_$　　・・・　式４

を使用する方がよい。但し，ここで記号「＿」は否定論理である。

【００３０】

その第１の理由は，式２では，３入力ＡＮＤＯＲ論理を取る必要があり，回路が複雑化するからである。そして第２の理由は，図１のマルチプレクサMPX_1（１０４）と第２のマルチプレクサMPX_2（１０５）のキャリ選択信号として排他論理和$P_k@Q_k$を用いることに関連するが，キャリセレクト方式加算器では，入力ＰやＱはキャリC_{in}より先に定まるので，マルチプレクサで選択する方式が速度的にも回路量的にも有利だからである。

【００３１】

図１の回路を実際にＬＳＩで実現する場合には，トランスミッションゲート（Ｔｒａｎｓｍｉｓｓｉｏｎ　Ｇａｔｅ）を用いれば回路量が削減できることが知られている。トランスミッションゲートを用いる論理（以下，「ＴＧ論理」という。）では，見かけ上ゲート数が少なくて済む。しかし，ドライブ能力を欠くことがあるので，設計上注意が必要である。例えば，必要に応じてインバータを追加し，若しくは負論理で動作させるなどの工夫が効果的である。

【００３２】

図２は，ＴＧ論理によるＸＯＲ論理回路であって，図１の素子XOR_1（１０２）で使用することができる。回路的には，２つのインバータと１つのＴＧから構成されるが，一方のインバータＩＮＶは，そのソース側が入力Q_k若しくは$Q_k_$に接続されるのが通常と異なる点である。図２で，基盤に矢印を持つトタンジスタがＰチャンネルＭＯＳであり，矢印のないものがＮチャンネルＭＯＳである。この様にＴＧ論理を使うとゲート数を削減できるのは，トランスミッションゲートとトグルスイッチとの類似性にあると考えられる。図２の回路では，式１

の前半部分，及び，式３，若しくは，式４の選択信号$P_k @ Q_k$を計算することができる。

【００３３】

図３は，ＴＧ論理によるＸＯＲ論理回路であって，図１の素子XOR_2（１０３）で使用することができる。回路的には，２つのインバータと２つのＴＧから構成される。キャリ選択信号$P_k @ Q_k$により，キャリC_k若しくはC_k＿が選択されるので，結果的に式１の加算値S_kを計算することができる。

【００３４】

図４は，ＴＧ論理によるマルチプレクサであって，図１のマルチプレクサMPX_1（１０４），または，マルチプレクサMPX_2（１０５）で使用することができる。回路的には，キャリ選択信号$P_k @ Q_k$によりキャリ入力C_{in1}若しくはキャリ入力C_{in2}かQ_kかを選択することにより，式４によるキャリ出力C_{out1}若しくはC_{out2}を計算することができる。入力をQ_kではなくP_kとすることにより，式３へ適用することもできる。

【００３５】

図５は，ＴＧ論理によるマルチプレクサであって，図１の第３のマルチプレクサMPX_3（１０１）で使用することができる。真のキャリC_sをキャリ選択信号として用いて，仮想キャリ入力C_{in1}若しくはC_{in2}のいずれかを真のキャリC_kとして選択することができる。

【００３６】

本発明の第１の実施の形態による図１の回路構成の加算器を，従来の回路構成の加算器と比較した場合に，単純にＴＧ論理で構成したとするときの対照表を表１に示す。単なる加算器は２０ゲートで構成できるで，ＴＧ論理の優秀性は明らかである。但し，配線容量で遅延量が大きく，ドライブ能力が欠けることがあるので設計上は注意が必要である。表１に示すように，従来の回路構成による加算器では４６ゲートを要したが，本発明の第１の実施の形態による図１の回路構成による加算器では，３２ゲートで済ませることができ，３割程度の回路量を削減することができた。また，従来の回路構成による４ビット加算器では１９０ゲートを要したが，本発明の第１の実施の形態の回路構成による４ビット加算器では

，１３４ゲートで済ませることができ，ここでも３割程度の回路量を削減することができた。

【００３７】

【表１】

	ＴＧ論理による回路量（ゲート数）
単なる加算器	２０
従来の方式	４６
本発明の方式	３２
従来の方式による４ビット加算器	１９０
本発明の方式による４ビット加算器	１３４

【００３８】

計算速度の面では，キャリ入力からキャリ出力に至るまでにマルチプレクサ１段分の遅延しか要しないことは重要である。この事実は，後述する本発明の第２の実施の形態の基礎を与える。キャリセレクト方式加算器では，入力ＰやＱはキャリC_{in}より先に定まるので，マルチプレクサのキャリ選択信号Ｐ＠Ｑは，キャリの伝搬に殆ど関与しない。ＴＧ論理によれば，わずか１段分のゲート遅延でキャリは１つの加算器を伝搬することができる。インバータを追加し，若しくは負論理で動作させたとしても，ゲート遅延２段分で十分である。

【００３９】

図１の本発明の第１の実施の形態によるキャリセレクト方式加算器の１ビット分の回路から，多ビットの加算器を容易に構成することができる。図６は，本発明の第１の実施の形態によるＲＣＳＡ（１００）を採用した，４ビットキャリセレクト方式加算器の構成図である。この加算器（２０４）は，第１の実施の形態によるＲＣＳＡ（２０１～２０４）が，４ビット分継続接続されている。キャリセレクト方式加算器では真のキャリC_{in}の値は’０’または’１’のいずれかであるので，仮想キャリを用いて加算を先に進めておき，最後に真のキャリ入力C_{in}で真のキャリ出力C_{out}，及び加算値S_0，S_2，またはS_3を選択す

る。このような構成では，加算値S_0，S_2，またはS_3の選択がそれらの加算後に行われるのではなく，選択されたキャリC_kを使う加算が行われるところが従来と異なる。これは，加算値の計算はキャリの生成と比べて遅くても良いと言う事情を利用している。

【００４０】

図６において，$RCSA_0$（２０１），$RCSA_1$（２０２），$RCSA_2$（２０３），及び$RCSA_3$（２０４）は，図１の構成を持つ本発明の実施の形態による簡易キャリセレクト方式加算器であり，$RCSA_0$～$RCSA_3$（２０１～２０４）は，順に継続接続されている。この加算器では，仮想キャリを定めて仮のキャリ出力C_{out1}若しくはC_{out2}を計算し，真のキャリ入力C_{in}であるキャリ選択信号C_sを用いて，マルチプレクサＭＰＸ（２０５）において真のキャリ出力C_{out}を選択する。なお，初段の加算器$RCSA_0$（２０１）において，図１の第３のマルチプレクサMPX_3（１０１）を省略し，直接真のキャリ入力C_{in}を選択後のキャリC_kとすることができる。また，仮想キャリVC_0は，値’０’，仮想キャリVC_1は値’１’と決めておくが，その逆であってもよい。VC_0とVC_1の値を逆にしたときはマルチプレクサＭＰＸ（２０５）において，上記と逆の選択をすればよい。

【００４１】

図７は，本発明の第１の実施の形態による４ビット加算器を採用した１６ビットキャリセレクト方式加算器の一実施形態である。この加算器は図７に示すように，図６の４ビットキャリセレクト方式加算器（４ＲＣＳＡ）を継続接続して，１６ビットキャリセレクト方式加算器を構成している。この実施形態では，継続する４ＲＣＳＡは３段のみで，初段は通常の４ビット加算器（３０１）を使用している。この実施形態は４ビット加算器による一種の並列演算であって，初段の加算器はキャリ選択回路が不要なことから回路量を減らす工夫をしている。従って，４ビット加算器（３０１）の方式は特に制限されず，通常のＣＰＡであっても，キャリルックアヘッド方式加算器，若しくは，キャリスキップ方式加算器であってもよい。

【００４２】

４ビット加算器（３０１）において，Ｘｂｕｓ（３０５）からの４ビット入力Ｐ３：０（０ビットから３ビットまでの入力を表す。以下同様。）と，Ｙｂｕｓ（３０６）からの４ビット入力Ｑ３：０とを，キャリ入力$Ｃ_{in}$を受けて加算し，加算値として４ビット出力Ｓ３：０と，キャリを出力する。この出力されたキャリは，次段の加算器$４ＲＣＳＡ_{1}$（３０２）のキャリ入力となる。次段の加算器$４ＲＣＳＡ_{1}$（３０２）は，図６のキャリセレクト方式加算器であって，Ｘｂｕｓ（３０５）からの４ビット入力Ｐ７：４と，Ｙｂｕｓ（３０６）からの４ビット入力Ｑ７：４とを，キャリ入力$Ｃ_{i}$を受けて加算し，加算値として４ビット出力Ｓ７：４と，キャリ$Ｃ_{0}$を出力する。このキャリ$Ｃ_{0}$は，３段目の加算器$４ＲＣＳＡ_{2}$（３０３）のキャリ入力となる。

【００４３】

３段目の加算器$４ＲＣＳＡ_{2}$（３０３）は，図６のキャリセレクト方式加算器であって，Ｘｂｕｓ（３０５）からの４ビット入力Ｐ１１：８と，Ｙｂｕｓ（３０６）からの４ビット入力Ｑ１１：８とを，キャリ入力$Ｃ_{i}$を受けて加算し，加算値として４ビット出力Ｓ１１：８と，キャリ$Ｃ_{0}$を出力する。このキャリ$Ｃ_{0}$は，３段目の加算器$４ＲＣＳＡ_{2}$（３０３）のキャリ入力となる。４段目の加算器$４ＲＣＳＡ_{3}$（３０４）は，図６のキャリセレクト方式加算器であって，Ｘｂｕｓ（３０５）からの４ビット入力Ｐ１５：１２と，Ｙｂｕｓ（３０６）からの４ビット入力Ｑ１５：１２とを，キャリ入力$Ｃ_{i}$を受けて加算し，加算値として４ビット出力Ｓ１５：１２と，キャリ出力$Ｃ_{out}$を出力する。

【００４４】

図８は，図７に示す本発明の実施形態によるキャリセレクト方式加算器を用いた１６ビット加算器の動作を説明するための，出力端子等の波形変化の形で遅延時間を表示したタイムチャートである。図８では，ＲＤ＿信号によってＸｂｕｓとＹｂｕｓのデータが確定し，加算結果が遅延時間Ｔｄの後，Ｚｂｕｓ上に現れるまでの変化により，本実施形態によるキャリセレクト方式加算器の高速性が示されている。図８で加算キャリの遅延時間をＴａとした場合，４ビット加算器$４ＲＣＳＡ_{0}$（３０１）の出力Ｓ３：０（０ビットから３ビットまでの出力を表す。以下同様。）は，遅延時間４Ｔａで加算が終了する。また，選択キャリのマル

チプレクサでの遅延時間をＴｓとした場合，加算器４ＲＣＳＡ$_1$（３０２）の出力Ｓ７：４は，遅延時間４Ｔａ＋Ｔｓで加算が終了する。また，加算器４ＲＣＳＡ$_2$（３０３）の出力Ｓ１１：８は，遅延時間４Ｔａ＋２Ｔｓで加算が終了する。また，加算器４ＲＣＳＡ$_3$（３０４）の出力Ｓ１５：１２は，遅延時間４Ｔａ＋３Ｔｓで加算が終了する。従って，加算結果がＺｂｕｓ上に現れるまでの遅延時間Ｔｄは，最大で４Ｔａ＋３Ｔｓ程度である。なお，通常の１６ビット加算器の出力の加算時間は，１６Ｔａである。

【００４５】

本実施形態のポイントは，データが確定した後に，４つの加算器（３０１～３０４）が仮想キャリによって同時に加算を始めることができる点にある。このようなことが可能になった結果，４ビット分の加算に要する時間はとられてしまうが，その後は選択キャリがマルチプレクサを通過する遅延時間Ｔｓだけで桁数の多い加算を実行することができるという利点を享受できる。本実施形態では，キャリは加算器上ではなく，マルチプレクサ上を伝搬して行くと捉えることもできる。

【００４６】

以上示したように，本発明の第１の実施の形態によれば，ＲＣＳＡの簡易なキャリセレクト方式加算器によって，従来のキャリセレクト方式加算器と比べて３割程度回路量を削減することができる。計算速度の面では，キャリ入力からキャリ出力に至るのにマルチプレクサ１段分の遅延しか要しない。

【００４７】

（第２の実施の形態）

本発明による第２の実施の形態は，本明細書で適応キャリセレクト方式（Ａｄａｐｔｉｖｅ　Ｃａｒｒｙ－ｓｅｌｅｃｔ　Ａｄｄｅｒ）と呼ぶ新しい方式に関する。この方式は，キャリ選択信号Ｃｓを用いるマルチプレクサ（例えば，図６のＭＰＸ）の遅延時間Ｔｓと，本実施形態によるキャリセレクト方式加算器のキャリ遅延時間Ｔａとが，ほぼ同等である事実に基礎をおいている。これは加算器を通過するキャリ（以下，加算キャリという。）が，図１の第１のマルチプレクサＭＰＸ$_1$または第２のマルチプレクサＭＰＸ$_2$の１段分の遅延（Ｔａ）を有し

，キャリ選択信号C_sが，図６のマルチプレクサＭＰＸの１段分の遅延（Ｔｓ）を有し，ＴａとＴｓがほぼ同等である。このことから，多ビットの加算において，次段のキャリセレクト方式加算器のビット数を増やしても，キャリ生成の遅延時間は変わらないことを利用している。この遅延時間は変わらないが，処理できるビット数は漸次増やしていけるので，全体の加算時間は短縮することになる。

【００４８】

図９は，本発明の第２の実施の形態による多ビット加算器の回路構成である。この加算器は，上述の適応キャリセレクト方式で構成したことを特徴とする。この加算器は，第１の実施の形態による加算器（A_0～A_n）が複数段継続接続されている。前段の第１のキャリ出力を後段の第１のキャリ入力とし，前段の第２のキャリ出力を後段の第２のキャリ入力とし，複数の加算器のキャリ選択入力を全段で共通とする。そして，初段の加算器の真のキャリ入力をキャリ選択入力とし，第１のキャリ入力を第１の仮想キャリとし，第２のキャリ入力を第２の仮想キャリとし，キャリ選択入力により最終段の加算器の第１と第２の２つのキャリ出力のいずれかを選択する第４のマルチプレクサを有する。そして，第４のマルチプレクサの出力をキャリ出力としている。なお，初段の加算器の第３のマルチプレクサは省かれ，初段の加算器の第３のマルチプレクサの出力の替わりにキャリ選択入力を用いている。

【００４９】

図９で，初段のM_0ビット加算器A_0は，第１の実施の形態によるＲＣＳＡである。そして，次段のM_1ビット加算器A_1～最終段のM_nビット加算器A_nも，全て第１の実施の形態によるＲＣＳＡで構成されている。

【００５０】

従って，Ｎビット加算器を対象とした場合，

$N=\Sigma M_k$　（$k=0, \cdots, n$）　　・・・　式５

が成立する。但し，最終段加算器A_nのビット数M_nは，必ずしも前段のビット数より大きいとは限らない。数Ｎが丁度良い値でなければ，通常はM_nは端数になるからである。

【００５１】

ここで，$M_1=M_0$のように構成すると，最初のマルチプレクサに加算器A_0とA_1のキャリをほぼ同時に入力することができる。これは入力側での最適化を意味する。また，最終段加算器の１つ手前の加算器A_{n-1}において，その出力側のマルチプレクサに加算器を経由したキャリとキャリセレクト信号が同時に入力することができれば，全体の加算時間（Ｔｔ）を最適化することができる。これは出力側での最適化を意味し，上記入力側での最適化と区別できる。最終段加算器A_nを対象にしなかったのは，そのビット数M_nが通常端数になるからである。

【００５２】

上記選択キャリ，即ち，加算器A_{n-1}に至るキャリセレクト信号C_sの遅れＴｄは，

$$Td=M_0Ta+(n-2)Ts \quad \cdots \text{式６}$$

で与えられる。他方，加算器A_{n-1}を経由した前記加算キャリの遅れが同じくＴｄであれば，出力側での最適化を達成することができ，

$$Td=M_{n-1}\cdot Ta \quad \cdots \text{式７}$$

で与えられる。なお，全体の加算時間Ｔｔは，

$$Tt=Td+2Ts \quad \cdots \text{式８}$$

である。

【００５３】

第２の実施の形態において，漸次増やした加算器のビット数を等差数列で構成してみる。すると，公差をｈビットとした場合，

$$M_k=M_0+h(k-1)\ (k=1,\cdots,n-1) \quad \cdots \text{式９}$$

と表現することができる。

【００５４】

式６及び式７より公差ｈを求めると，

$$h=Ts/Ta \quad \cdots \text{式１０}$$

となる。そして，式５及び式９よりｎを求めると，

$$n=\{-(2M_0-3h)+\sqrt{\{(2M_0-3h)^2+8h(N-M_n-h)\}}\}/2h \quad \cdots \text{式１１}$$

となる。

【００５５】

また，全体の加算時間Ｔｔを最小化するために初期値M_0をどのように定めるのが良いかを計算すると，

$M_0 = 3h/2$　　・・・　式１２

となり，このときキャリを選択するマルチプレクサの個数ｎは，

$n = \sqrt{\{2(N - M_n - h)/h\}}$　　・・・　式１３

と求まる。そして，Ｔｔの最小値（Ｔｔ）ｍｉｎは，式１３のｎを用いて，

$(Tt)min = (n + 3/2)h \cdot Ta$　　・・・　式１４

で与えられる。

【００５６】

公差ｈは式１０で与えられるので，上記考察に基づき１に近い値であると推定できる。第１の実施の形態によるＲＣＳＡを採用した場合には，選択キャリのファンアウトが多いので，加算キャリの遅延時間Ｔａよりも選択キャリの遅延時間Ｔｓの方が大きくなることがある。従って，公差ｈは１に近く，かつ，１より大きい数になることがある。しかし，第１の実施の形態によるＲＣＳＡを採用しない場合は，加算キャリが選択キャリよりも遅いのが通常で，従ってｈは通常１よりも小さくなる。

【００５７】

実際の回路では必ずしも全体の加算時間Ｔｔを最小化するように上記変数を定めることはできない。変数ｈは，１とか，２とか，０．５とか，整数若しくは整数分の１を採用しなければならず，また，初段の加算器A_0のビット数M_0も整数でなければならない。従って，上記最適化に沿った最善の設計が必要になる。

【００５８】

図１０，図１１は，上記観点から，多ビット加算器での本実施形態による具体的構成を例示したものである。図１０（ａ）は，６４ビット加算器及び１２８ビット加算器について従来の構成法を例示してある。ここでは単純に４ビット加算器を継続接続し，その都度真のキャリを選択する。この方式では，選択キャリの遅延時間が単純に加算されていくので無駄が多い。

【００５９】

図１０（ｂ）は，本発明の第２の実施の形態であって，初期値M_0＝１で，公差ｈ＝１の場合である。初期値M_0＝１であることから，初段の加算器A_0は１ビット，次段の加算器A_1も１ビットである。以下，公差１で加算器のビット数が増加する構成を採用する。６４ビット加算器の場合（Ｎ＝６４），ｎ＝１０．７であるので，A_{10}は１０ビット加算器で，最後の加算器A_{11}は端数ビット，M_n＝８になる。６４ビット加算器の場合（Ｎ＝１２８），ｎ＝１５．４であるので，A_{15}は１５ビット加算器で，最後の加算器Ａ１６は端数ビット，M_n＝７になる。M_nを最初から定めることができないので，実際には式１１からｎを求めることはできず，およそのｎの値しか知ることはできない。実際にM_nを定めるには，ｎを求めた後，式５により再計算することになる。

【００６０】

図１０（ｃ）は，本発明の第２の実施の形態であって，初期値M_0＝１で，公差ｈ＝０．５の場合である。公差ｈ＝０．５を実現するために加算器のビット数を加算器２個毎に１ビットづつ増加させている。公差が１／２の場合は，３番目の加算器A_2の出力側のマルチプレクサにおいて，選択キャリよりも加算器A_2の加算キャリの方が遅くなり，この事情は後段の加算器でも同様である。そこで，全体の計算時間Ｔｔを求めるに際し，端数M_nが出る場合は加算器A_{n-1}の遅延時間に基づいて計算し，その値はM_{n-1}・Ｔａ＋２Ｔｓである。幸いにして，キャリセレクト方式加算器においては遅延時間の累積は起こらない。即ち，一種の並列計算であるので，前段の計算で生じた遅延が後段の計算時間を積み増すことはない。

【００６１】

図１１（ｄ）は，本発明の第２の実施の形態であって，初期値M_0＝１で，公差ｈ＝２の場合である。公差ｈ＝２，即ち選択キャリの方が加算キャリよりも倍も遅いという事態は，本実施の形態の構成を採用する限り，余り起こらない。但し，キャリ選択信号C_Sのファンアウトが大きく，長い配線が想定されるので，全く起こらないとも言い切れない。ここでは比較のために例示した。

【００６２】

図１１（ｅ）は，本発明の第２の実施の形態であって，初期値M_0＝２で，公差ｈ＝１の場合である。上述の如く，全加算時間Ｔｔを最小化するための初期値M_0の値は式１２で与えられ，公差ｈ＝１の場合M_0＝１．５とするのが最適である。しかし，M_0は整数を選ばなければならないので，初期値M_0＝２とする設計を採用することもできる。図１０（ｂ）のM_0＝１の場合と比べると，全加算時間Ｔｔは殆ど同じである。図１１（ｆ）や図１１（ｇ）でも同様のことが言える。表２では，全加算時間Ｔｔに関し，例えばｈ＝１の場合，理論上の最適値と最善の設計の値との差は１％以下であり，この様な設計の有効性を示している。

【００６３】

表２に，図１０，図１１で掲げた本発明の第２の実施の形態について，全加算時間Ｔｔを計算した値を列挙した。図１０（ａ）の従来例と比べると，６４ビット加算器で４割程度，１２８ビット加算器で５割程度，全加算時間Ｔｔを減らすことができる。勿論通常のＣＰＡと比べると格段に速い。全加算時間Ｔｔは最終段の１つ手前の加算器A_{n-1}の加算キャリの遅延時間程度であり，より正確には，ｈ＝１の場合ＣＰＡの全加算時間との比はおよそ，　$N/(M_{n-1}-2)$

である。この値は６４ビット加算器で８倍，１２８ビット加算器で１０倍程度である。

【００６４】

【表２】

加算器の構成方法	M_0	h	全加算時間　Ｔｔ　（Ｔｔ/Ｔａ）		備考
			６４ビット	１２８ビット	
図１０（ａ）	４	０	４Ｔａ＋１５Ｔｓ ――	４Ｔａ＋３１Ｔｓ ――	従来例
図１０（ｃ）	１	1/2	８Ｔａ＋　Ｔｓ （８．５）	１１Ｔａ＋２Ｔｓ （１２）	本発明例
図１０（ｂ）	１	１	Ｔａ＋１１Ｔｓ （１２）	Ｔａ＋１６Ｔｓ （１７）	〃
図１１（ｄ）	１	２	Ｔａ＋　８Ｔｓ （１７）	Ｔａ＋１２Ｔｓ （２５）	〃
図１１（ｆ）	２	1/2	８Ｔａ＋　２Ｔｓ （　９）	１１Ｔａ＋２Ｔｓ （１２）	〃
図１１（ｅ）	２	１	２Ｔａ＋１０Ｔｓ （１２）	２Ｔａ＋１５Ｔｓ （１７）	〃
図１１（ｇ）	２	２	２Ｔａ＋　８Ｔｓ （１８）	２Ｔａ＋１１Ｔｓ （２４）	〃
理論上の最適値	3/2 h	h	１．５Ｔａ＋１０．５Ｔｓ （１２．０）	１．５Ｔａ＋１５．５Ｔｓ （１７．０）	ｈ＝１と仮定

【００６５】

表２において，公差ｈが１／２の場合は，３番目の加算器A_2の出力側のマルチプレクサにおいて，選択キャリよりも加算器A_2の加算キャリの方が遅くなる。この事情は後段の加算器でも同様である。

【００６６】

回路量でみた場合，図１０（ａ）の従来例と比べると，加算器の総ビット数Ｎは同じであるが，キャリの選択に要するマルチプレクサの数を減らせることができる。具体的には，６４ビット加算器で４個程度，１２８ビット加算器で１５個程度減らせる。

【００６７】

以上示したように，本発明の第２の実施の形態によれば，本明細書で適応キャリセレクト方式と呼ぶ新しい方式を採用し，多ビット値の高速加算が達成され，併せて回路量の削減をすることができる。

【００６８】

（第３の実施の形態）

図１２は，本発明の第２の実施形態の適応キャリセレクト方式を，３２ビットブース乗算器（Ｂｏｏｔｈ　Ｍｕｌｔｉｐｌｉｅｒ）に応用した第３の実施の形態の構成を示すブロック回路図である。図１２では，２次ブース乗算器のワレスツリー（Ｗａｌｌａｃｅ　Ｔｒｅｅ）を，ＣＳＡ（Ｃａｒｒｙ　Ｓａｖｅ　Ａｄｄｅｒ）で構成している。$Ｐ_{16}$～$Ｐ_{0}$の１７個の値は，２次ブース乗算器の部分積であって，それぞれ３３ビットで構成される。３－２ＣＳＡ（１４８）は，３つの数値を２つにまとめることができるＣＳＡであり，これによって加算に要する数値の削減を行う。また，４－２ＣＳＡ（１４１，１４２，１４３，１４４，１４５，１４６，１４７）は，４つの数値を２つにまとめることができるＣＳＡであり，３－２ＣＳＡを継続接続して構成する。

【００６９】

通常の２次ブース乗算器のワレスツリーでは，最終段加算器の位置にはＡＣＳＡ（１４９）ではなく，ＣＰＡ（Ｃａｒｒｙ　Ｐｒｏｐａｇａｔｉｏｎ　Ａｄｄｅｒ）が用いられ，２つの数値の加算が行われる。このＣＰＡは最終的に残った２つの６４ビット値を加算するためのもので，６５ビット構成であり，唯一キャリの伝搬で大きな遅延が発生し，乗算器の高速化のネックとなっていた部分である。第３の実施形態は，適応キャリセレクト方式加算器（ＡＣＳＡ）をこのＣＰＡの替わりに採用したもので，３２ビット乗算器の高速化に著しい寄与をなしている。なお，２の補数表示により３２ビット値の先頭ビット（ＭＳＢ）を符号にした場合も同様である。

【００７０】

将来の６４ビット乗算器等を視野に入れた場合に，多ビット値の高速加算は必須の技術に成りつつあり，本発明の第３の実施の形態によるＡＣＳＡを採用した高速乗算器は，単なる構成要素の置き換えに留まらない効果を発揮する。例えば，６４ビット乗算器を２次のブース乗算器で構成した場合，上記ＡＣＳＡ（１４９）は，３２ビット乗算器の場合のおよそ２倍の多ビット値を取り扱うことになる。これを従来通りのＣＰＡで構成した場合，そのキャリの遅延時間は乗算器の

高速化の大きな障害となるであろうし，６４ビット乗算器そのものの採用意義を失わせることにもなりかねない。

【００７１】

以上示したように，本発明の第３の実施の形態によれば，２次ブース乗算器のワレスツリーへの応用により，乗算器の高速化に著しく寄与することができる。

【００７２】

なお，上記説明した実施形態の加算器，乗算器は，単独ではもちろん，集積回路の一部として組み込まれてもその効果を発揮することはいうまでもない。

【００７３】

以上，添付図面を参照しながら本発明の加算器，乗算器，及びこれを用いた集積回路の好適な実施形態について説明したが，本発明はこれらの例に限定されない。いわゆる当業者であれば，特許請求の範囲に記載された技術的思想の範疇内において各種の変更例または修正例に想到し得ることは明らかであり，それらについても当然に本発明の技術的範囲に属するものと了解される。

【００７４】

【発明の効果】

本発明により，従来のキャリセレクト方式加算器を改良して，多ビット乗算器や暗号技術などにおける多ビットの加算に最適な，回路量を減らし，より高速とした加算器，乗算器の回路構成が提供できた。

【図面の簡単な説明】

【図１】

図１は，本発明の第１の実施の形態によるキャリセレクト方式加算器の１ビット分の構成を示すブロック回路図である。

【図２】

図２は，ＴＧ論理によるＸＯＲ論理回路である。

【図３】

図３は，ＴＧ論理によるＸＯＲ論理回路である。

【図４】

図４は，ＴＧ論理によるマルチプレクサである。

【図５】

図５は，ＴＧ論理によるマルチプレクサである。

【図６】

図６は，本発明の第１の実施の形態によるＲＣＳＡを採用した４ビットキャリセレクト方式加算器の構成図である。

【図７】

図７は，本発明の第１の実施形態による４ビット加算器を採用した１６ビットキャリセレクト方式加算器の回路構成図である。

【図８】

図８は，図７に示すキャリセレクト方式加算器を用いた１６ビット加算器の動作を説明するための，出力端子等の波形変化の形で遅延時間を表示したタイムチャートである。

【図９】

図９は，本発明の第２の実施の形態による多ビット加算器の回路構成図である。

【図１０】

図１０は，（ａ）従来の構成と，（ｂ），（ｃ）本発明の第２の実施の形態による多ビット加算器の具体的構成を例示した回路図である。

【図１１】

図１１（ｄ）～（ｇ）は，本発明の第２の実施の形態による多ビット加算器の具体的構成を例示した回路図である。

【図１２】

図１２は，本発明の第３の実施の形態による３２ビットブース乗算器の一例を示す構成図である。

【符号の説明】

１００　ＲＣＳＡ

P_k　第１のデータ入力

Q_k　第２のデータ入力

C_{in1}　第１のキャリ入力

C_{in2}　第２のキャリ入力

C_s　キャリ選択入力

C_{out}　キャリ出力

１０１　第３のマルチプレクサ（MPX_3）

１０２　第１のＸＯＲ素子（XOR_1）

１０３　第２のＸＯＲ素子（XOR_2）

１０４　第１のマルチプレクサ（MPX_1）

１０５　第２のマルチプレクサ（MPX_2）

２０１　$RCSA_0$

２０２　$RCSA_1$

２０３　$RCSA_2$

２０４　$RCSA_3$

２０５　マルチプレクサ

Ｓ　加算値

C_{out1}，C_{out2}　仮のキャリ出力

３０１　通常の４ビット加算器

３０２　加算器４ＲＣＳＡ１

３０３　加算器４ＲＣＳＡ２

３０４　加算器４ＲＣＳＡ３

３０５　Ｘｂｕｓ

３０６　Ｙｂｕｓ

３０７　Ｚｂｕｓ

C_0　キャリ

A_0～A_n　加算器

４００　４ビットキャリセレクト方式加算器

４０１，４０２　４ビット加算器

４０３，４０４　マルチプレクサ

５００　１ビット分の回路

５０１，５０２　全加算器

５０３　マルチプレクサ

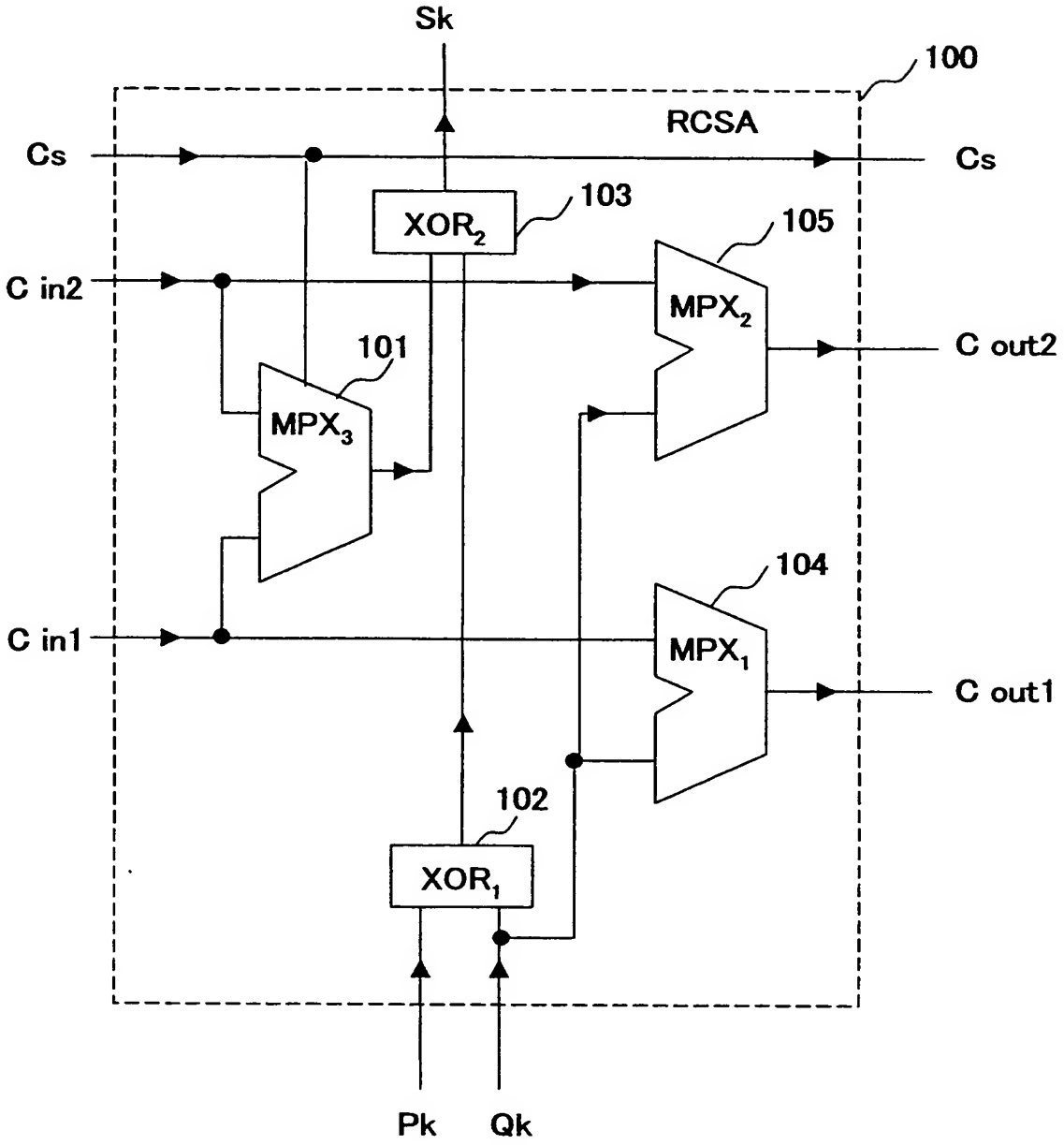
１４１～１４７　４－２ＣＳＡ

１４８　３－２ＣＳＡ

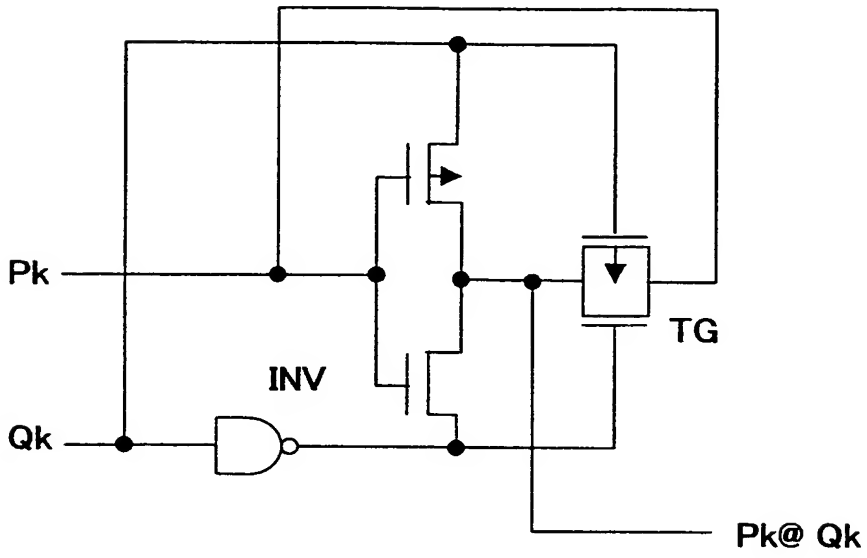
１４９　ＡＣＳＡ

【書類名】　図面

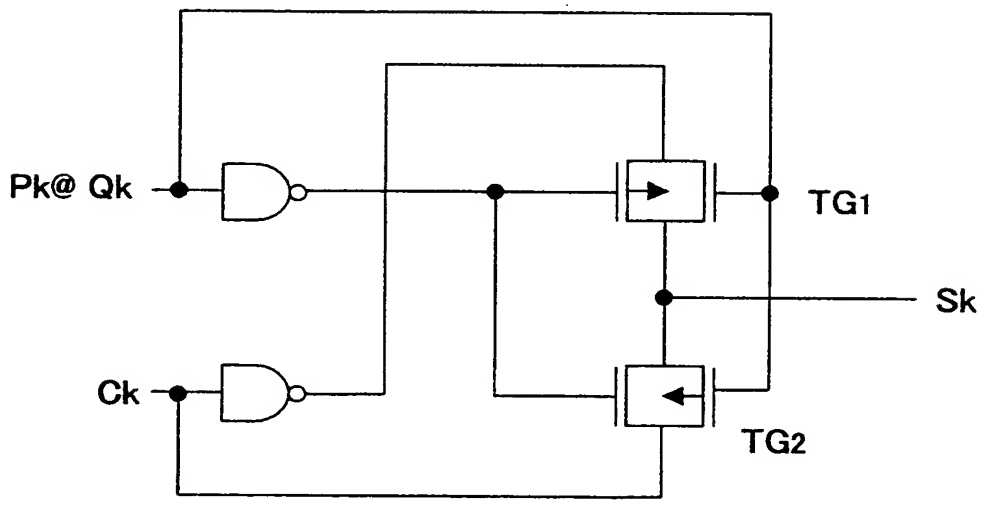
【図１】


Sk
100
RCSA
Cs
Cs
103
XOR2
105
C in2
MPX2
C out2
101
MPX3
104
C in1
MPX1
C out1
102
XOR1
Pk
Qk

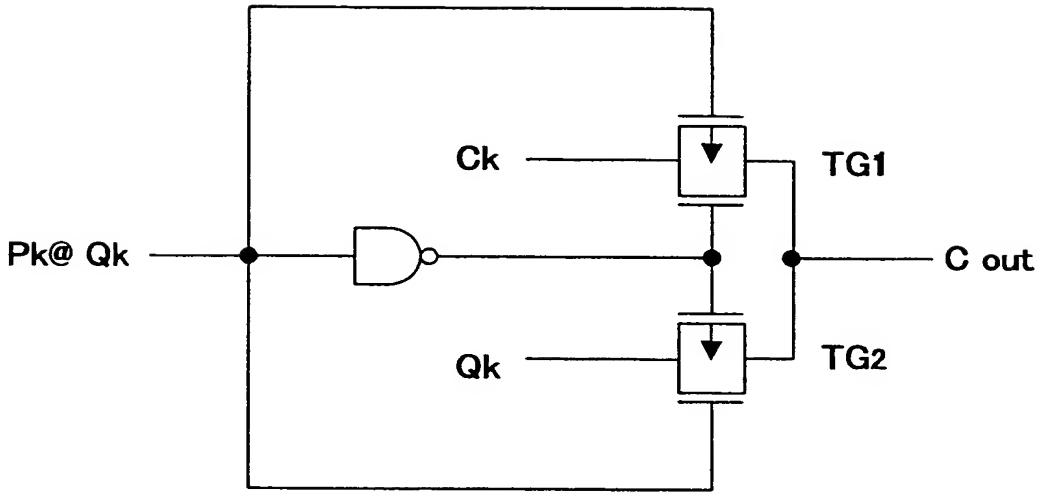
【図２】



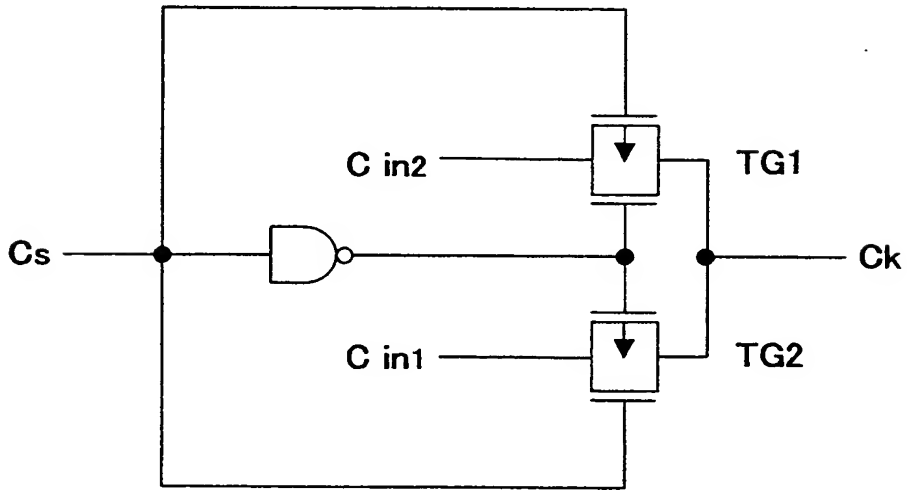
【図３】



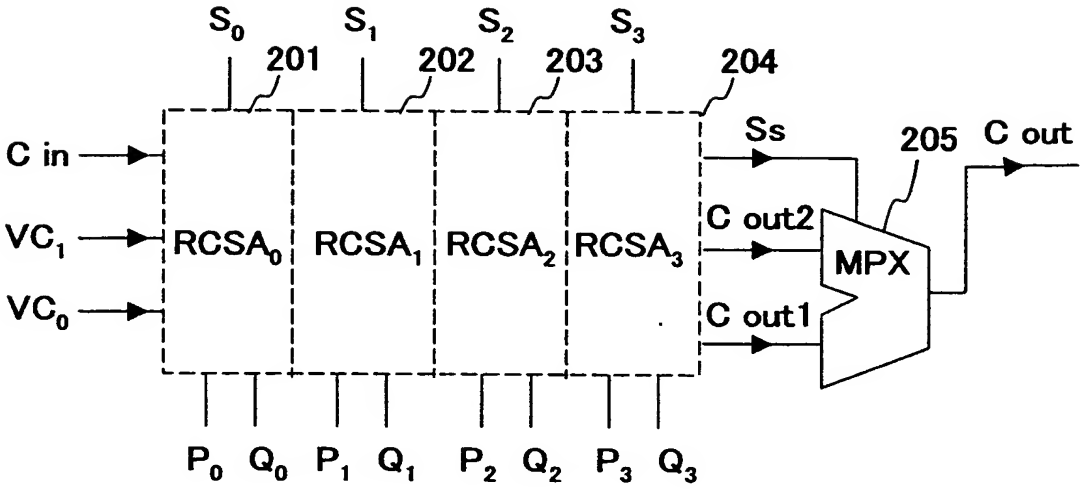
【図４】



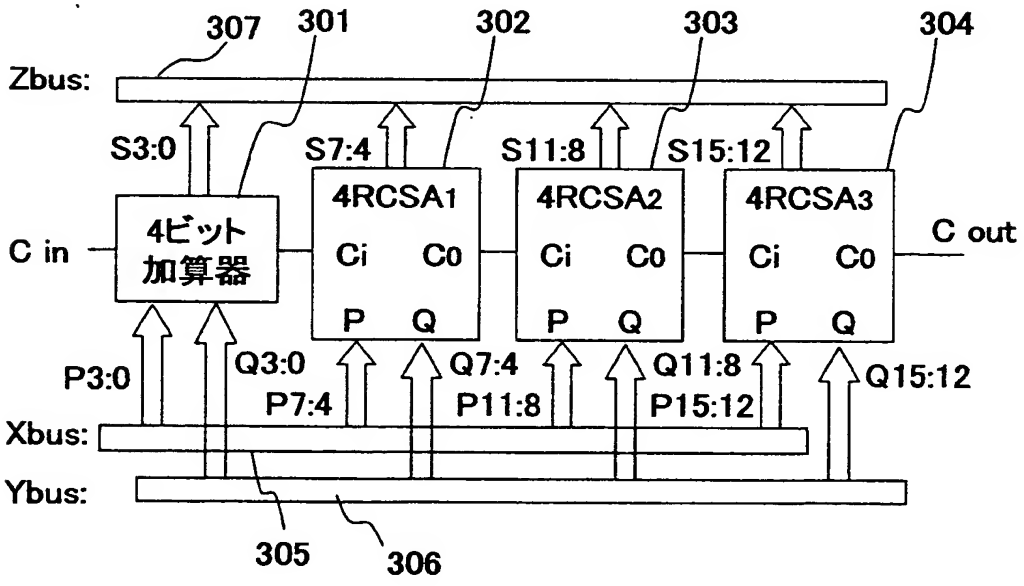
【図５】



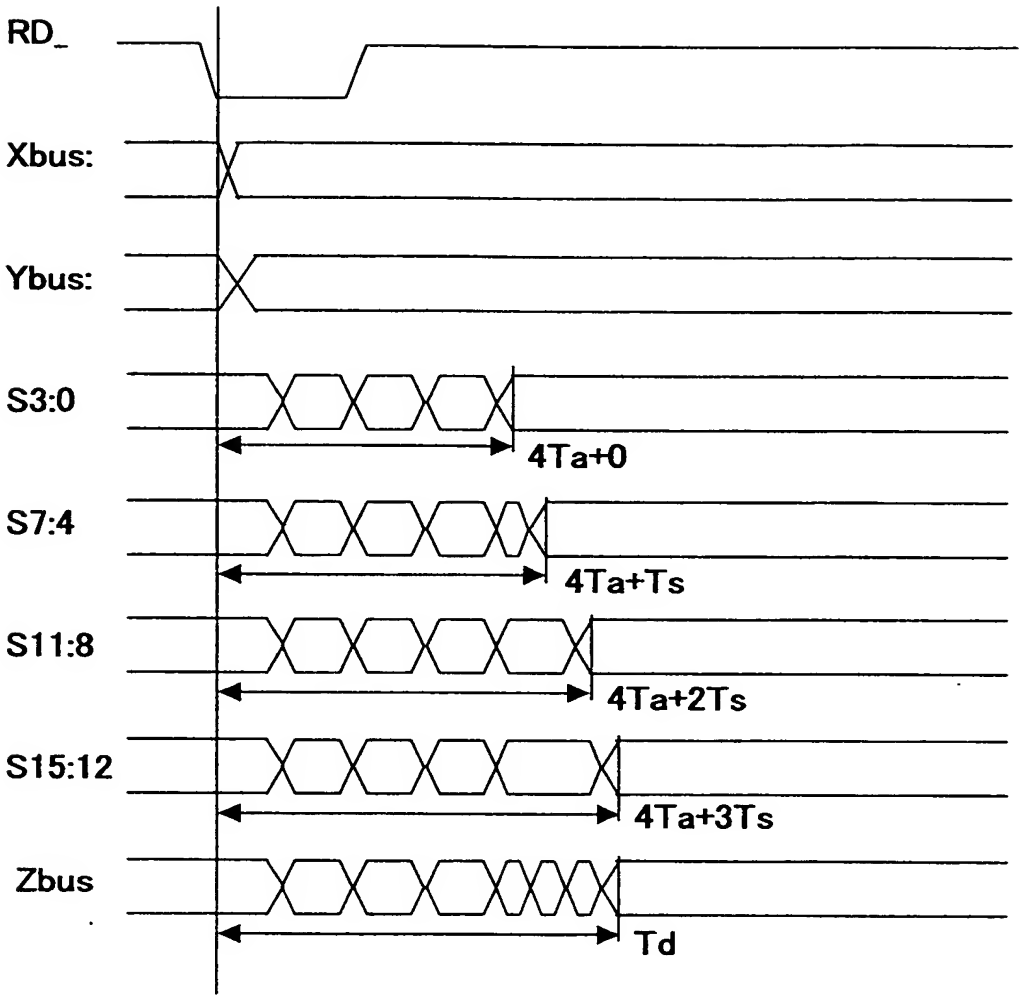
【図６】


S0
S1
S2
S3
201
202
203
204
C in
VC1
VC0
RCSA0
RCSA1
RCSA2
RCSA3
Ss
205
C out
C out2
MPX
C out1
P0
Q0
P1
Q1
P2
Q2
P3
Q3

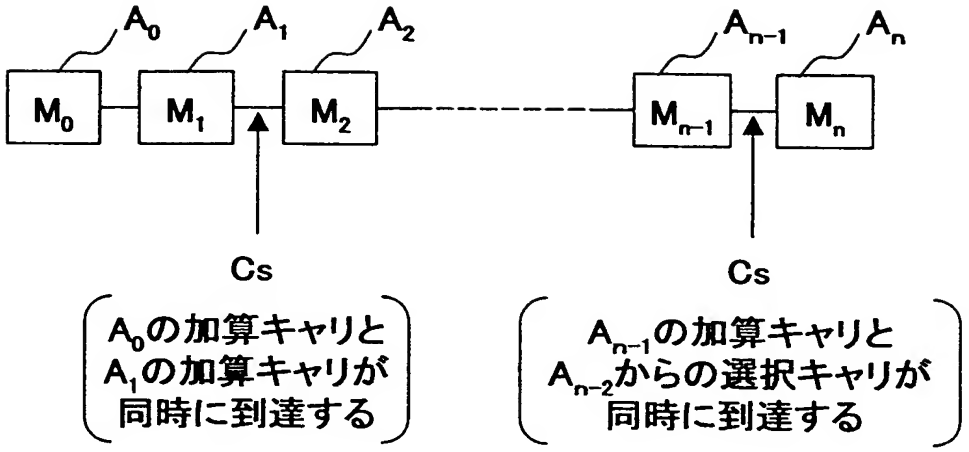
【図７】


307
301
302
303
304
Zbus:
S3:0
S7:4
S11:8
S15:12
C in
4ビット
加算器
4RCSA1
4RCSA2
4RCSA3
Ci
Co
P
Q
C out
P3:0
Q3:0
P7:4
Q7:4
P11:8
Q11:8
P15:12
Q15:12
Xbus:
Ybus:
305
306

【図８】


RD_
Xbus:
Ybus:
S3:0
4Ta+0
S7:4
4Ta+Ts
S11:8
4Ta+2Ts
S15:12
4Ta+3Ts
Zbus
Td

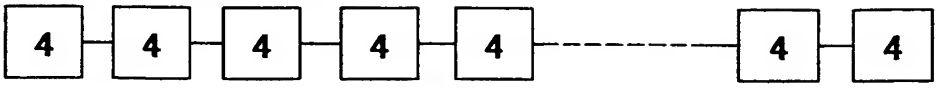
【図９】


A0
A1
A2
An-1
An
M0
M1
M2
Mn-1
Mn
Cs
Cs
A0の加算キャリと
A1の加算キャリが
同時に到達する
An-1の加算キャリと
An-2からの選択キャリが
同時に到達する

【図１０】

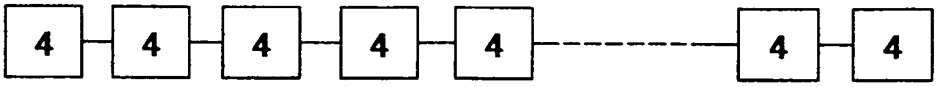
(a)従来の構成

64ビット加算器の場合



4RCSAを16個継続接続する。

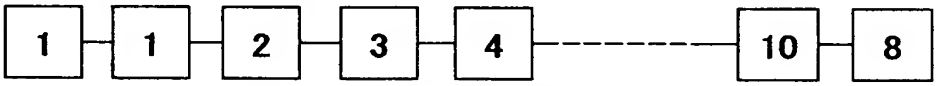
128ビット加算器の場合



4RCSAを32個継続接続する。

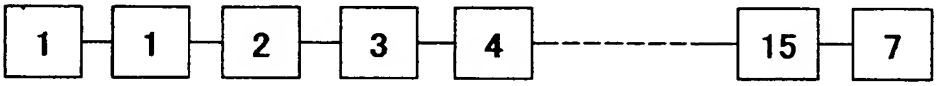
(b)初期値 M0＝1、公差 h＝1 の場合

64ビット加算器の場合



RCSAのビット数が漸次増加する。

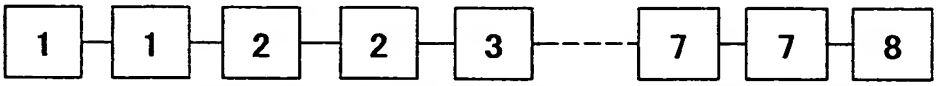
128ビット加算器の場合



RCSAのビット数が漸次増加する。

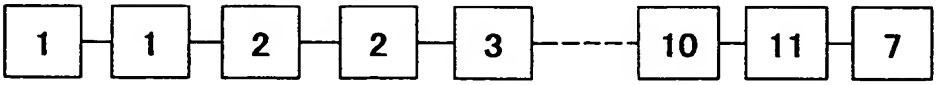
(c)初期値 M0＝1、公差 h＝0.5 の場合

64ビット加算器の場合



RCSAのビット数が2個おきに漸次増加する。

128ビット加算器の場合

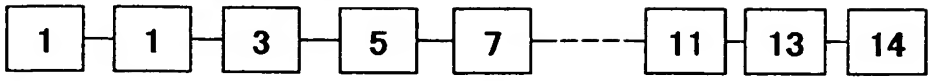


RCSAのビット数が2個おきに漸次増加する。

【図１１】

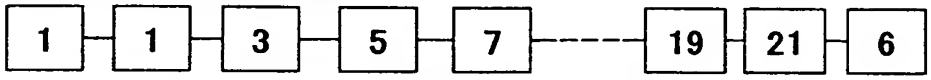
(d)初期値 M0＝1、公差 h＝2 の場合

64ビット加算器の場合



RCSAのビット数が2個づつ漸次増加する。

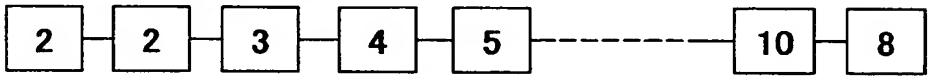
128ビット加算器の場合



RCSAのビット数が2個づつ漸次増加する。

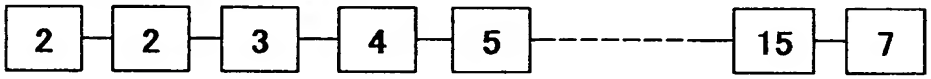
(e)初期値 M0＝2、公差 h＝1 の場合

64ビット加算器の場合



RCSAのビット数が漸次増加する。

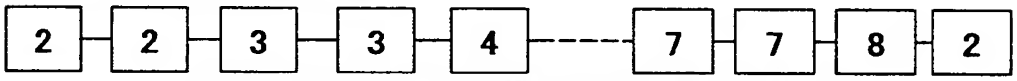
128ビット加算器の場合



RCSAのビット数が漸次増加する。

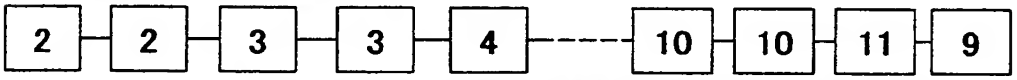
(f)初期値 M0＝2、公差 h＝0.5 の場合

64ビット加算器の場合



RCSAのビット数が2個おきに漸次増加する。

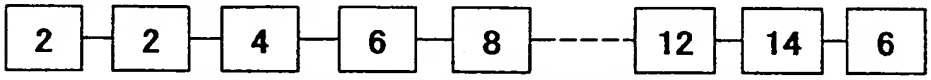
128ビット加算器の場合



RCSAのビット数が2個おきに漸次増加する。

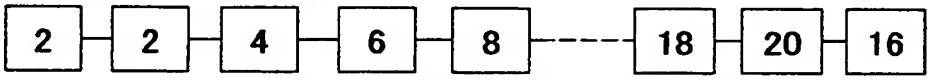
(g)初期値 M0＝2、公差 h＝2の場合

64ビット加算器の場合



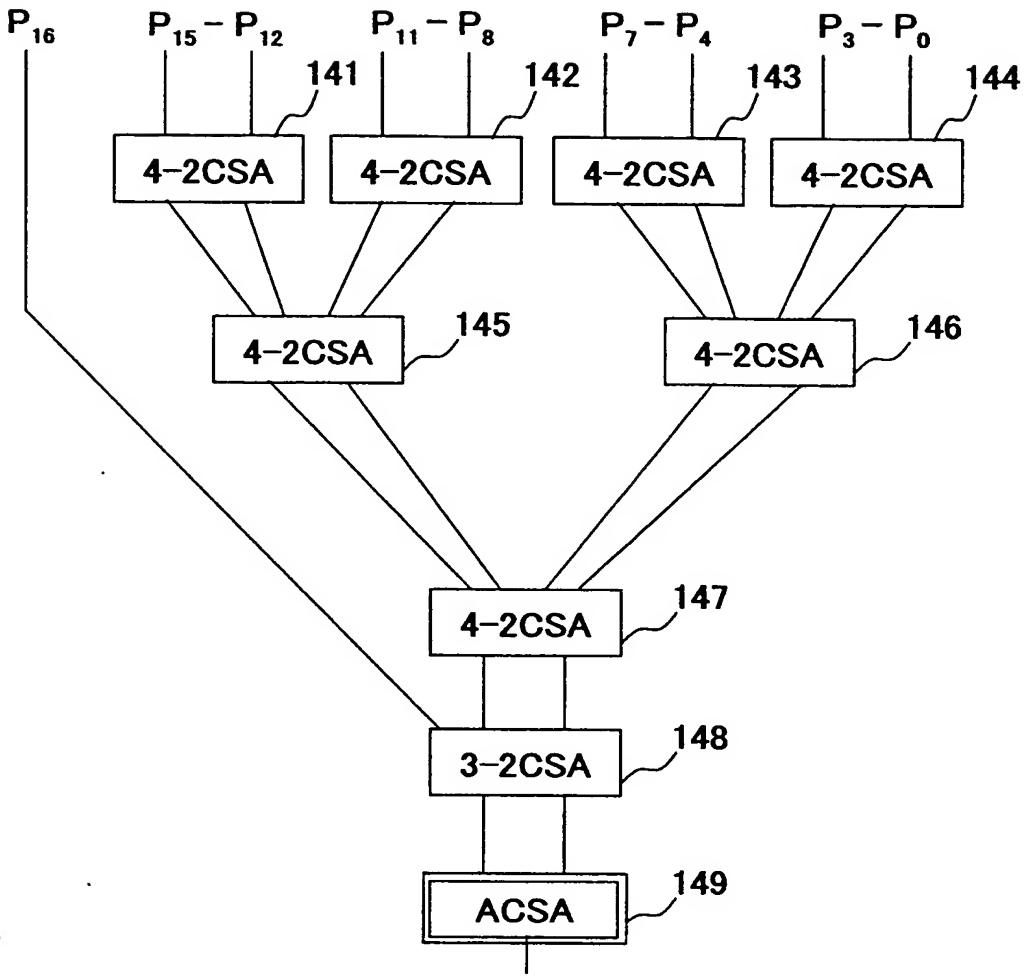
RCSAのビット数が2個づつ漸次増加する。

128ビット加算器の場合



RCSAのビット数が2個づつ漸次増加する。

【図１２】


P16
P15 – P12
P11 – P8
P7 – P4
P3 – P0
141
142
143
144
4–2CSA
4–2CSA
4–2CSA
4–2CSA
145
146
4–2CSA
4–2CSA
147
4–2CSA
148
3–2CSA
149
ACSA

【書類名】　要約書

【要約】

【課題】　暗号技術などにおける多ビットの加算に最適な，回路量を減らし，より高速としたキャリセレクト方式加算器加算器を提供する。

【解決手段】　２つのデータ入力と，２つのキャリ入力と，キャリ選択入力とを入力とし，２つのデータ入力のＸＯＲ出力を生成する第１ＸＯＲ素子と，このＸＯＲ出力を選択信号として第１キャリ入力と第１データ入力のいずれかを選択する第１マルチプレクサと，第２キャリ入力と第２データ入力のいずれかを選択する第２マルチプレクサと，キャリ選択入力を選択信号として２つのキャリ入力のいずれかを選択する第３マルチプレクサと，第３マルチプレクサの出力とＸＯＲ出力とのＸＯＲ出力を生成する第２ＸＯＲ素子とを有し，第１マルチプレクサの出力を第１キャリ出力とし，第２マルチプレクサの出力を第２キャリ出力とし，第３マルチプレクサの出力を加算値とする加算器。

【選択図】　図１

出　願　人　履　歴　情　報

識別番号　　〔０００００２９５〕

１．変更年月日　　１９９０年　８月２２日
［変更理由］　新規登録
住　所　　東京都港区虎ノ門１丁目７番１２号
氏　名　　沖電気工業株式会社